

REDUCING COSTS OF MANAGING AND ACCESSING NAVIGATION AND ANCILLARY DATA BY RELYING ON THE EXTENSIVE CAPABILITIES OF NASA'S SPICE SYSTEM

Boris V. Semenov⁽¹⁾, Charles H. Acton, Jr.⁽¹⁾, Nathaniel J. Bachman⁽¹⁾, Lee S. Elson⁽¹⁾, Edward D. Wright⁽¹⁾

⁽¹⁾ Jet Propulsion Laboratory, 4800 Oak Grove Dr., Pasadena, CA, 91109, USA, Email: Name.Surname@jpl.nasa.gov

ABSTRACT

The SPICE system of navigation and ancillary data possesses a number of traits that make its use in modern space missions of all types highly cost efficient. The core of the system is a software library providing API interfaces for storing and retrieving such data as trajectories, orientations, time conversions, and instrument geometry parameters. Applications used at any stage of a mission life cycle can call SPICE APIs to access this data and compute geometric quantities required for observation planning, engineering assessment and science data analysis. SPICE is implemented in three different languages, supported on 20+ computer environments, and distributed with complete source code and documentation. It includes capabilities that are extensively tested by everyday use in many active projects and are applicable to all types of space missions – flyby, orbiters, observatories, landers and rovers. While a customer's initial SPICE adaptation for the first mission or experiment requires a modest effort, this initial effort pays off because adaptation for subsequent missions/experiments is just a small fraction of the initial investment, with the majority of tools based on SPICE requiring no or very minor changes.

1. Introduction

SPICE is a system of navigation and ancillary data developed under the direction of NASA's Science Directorate by the Navigation and Ancillary Information Facility (NAIF) group at the Jet Propulsion Laboratory. The purpose of SPICE is to assist scientists in planning observations for, and analyzing data from space-borne instruments, and to assist engineers involved in modelling, planning and executing activities needed to conduct space exploration missions. The two primary parts of SPICE are data files, called "kernels", containing various navigation and other ancillary information, structured and formatted for easy access and correct use, and software, called the SPICE Toolkit, used to access the data and calculate various observation geometry parameters of interest.

As a concept of carefully archiving the fundamental navigation and ancillary data sets needed to derive observation geometry parameters, SPICE was first proposed in 1983 by NASA's Planetary Data Workshop. This proposal followed the recommendation for archival

treatment of data made by the National Research Council's Committee on Data Management and Computation (CODMAC) [1]. A year later, in 1984, as the concept was refined during the development of the Pilot Planetary Data System, the SPICE acronym, identifying the major system components, was introduced. To prove the concept a set of data formats and related software – a pre-cursor to SPICE – was implemented and used to support the Voyager Neptune and Uranus flybys. Successful demonstration on Voyager gave a green light to the development of the current SPICE system, which started in 1989. As the system matured, Magellan became the first mission that officially used SPICE in operations by generating and distributing spacecraft trajectory data in SPICE SPK format. Following successful use of SPICE on Magellan, both Mars Observer and Galileo chose SPICE as the replacement for the Supplemental Experiment Data Records (SEDR) system previously used for providing and archiving navigation data and ancillary on earlier U.S. planetary missions. Over the next decade SPICE became an integral part of the ground operations systems of, and the defacto standard for navigation and ancillary data in NASA's solar system exploration missions. As SPICE capabilities and recognition increased, various space agencies outside of the U.S. became interested in using it. In 2000 SPICE was proposed and accepted by the European Mars Express project as a supplementary (non-official) mechanism for distribution of navigation data in support of science data analysis and archiving. Later in 2004 SPICE was adopted for the same purpose by three more European Space Agency missions – Rosetta, Venus Express and SMART-1. In the past two years SPICE had also been deployed in support of science operations by the Japan Space Exploration Agency's (JAXA) on the Hayabusa and SELENE missions.

2. SPICE data

The first major part of the SPICE system is data. Different information types in SPICE are grouped together into logical elements and stored in underlying physical data files called "kernels" (see Fig. 1). Five kernels – SPK, PCK, IK, CK, and EK, – whose first letters make up the SPICE acronym, are the "cornerstones" of SPICE. SPK (Spacecraft Planet Kernel) deals with position information such as planet,

satellite, comet and asteroid ephemerides, spacecraft trajectories, tracking station locations, landing sites, rover path, and relative locations of science instruments or instrument detectors. PCK (Planetary Constants Kernel) provides parameters of the natural bodies including data needed to compute orientation of the body-fixed frames, using either IAU or high precision (for Earth and Moon) models, as well as parameters defining natural body shapes and sizes. IK (Instrument Kernel) serves as a repository for science instrument parameters needed to compute observation geometry, including field-of-view definition, detector geometry and timing offset specifications. CK (C-matrix Kernel) deals with orientation data for spacecraft and moving instruments or instrument parts. EK (Events Kernel) provides various types of mission and spacecraft event information such observations plans, records of executed commands, and notes describing how operations and observations were carried out.

In addition to the main five kernels, SPICE includes a few extra elements and corresponding supporting data files. Two of them – Leapseconds Kernel (LSK) and Spacecraft Clock Kernel (SCLK) – contain data needed to support conversions between the three main time systems supported in SPICE: UTC, ephemeris time, and spacecraft on-board time. The other critical element, Frames Kernel (FK), defines reference frames, associates them with orientation data provided in other kernels, and establishes connections between them.

3. SPICE Software

The second second major part of the SPICE system is software. The core of the SPICE software is a library that provides API interfaces for accessing data from various kernels as well for computing a large number of derived space geometry quantities using this data. User-developed applications call APIs provided in this library to compute and manipulate position and velocity vectors and reference frame transformations, calculate sub-spacecraft and surface intercept points and illumination conditions at these points, perform time conversions, as well as to access many other geometry functions. While the library includes over 1,000 functions, its hierarchical architecture makes it possible to utilize only a handful of the highest-level APIs to access and combine data from all SPICE kernels provided to a user-developed SPICE based applications. For example, if sufficient data is available, the two highest level APIs providing access to trajectory data can compute relative position or state vectors, either geometric or corrected for light time and/or stellar aberration, of any spacecraft or natural bodies, tracking stations or other surface locations given in any of a large number of references frames including inertial, body-fixed, topocentric, spacecraft, science instrument, spacecraft structure frames as well as frames defined using dynamic (time varying) basis vectors.

One of the main reasons why use of SPICE by planetary mission engineering and science communities has

Navigation and Ancillary Information ... stored in ... SPICE Data Files, "Kernels", ... accessed via ... SPICE Toolkit APIs

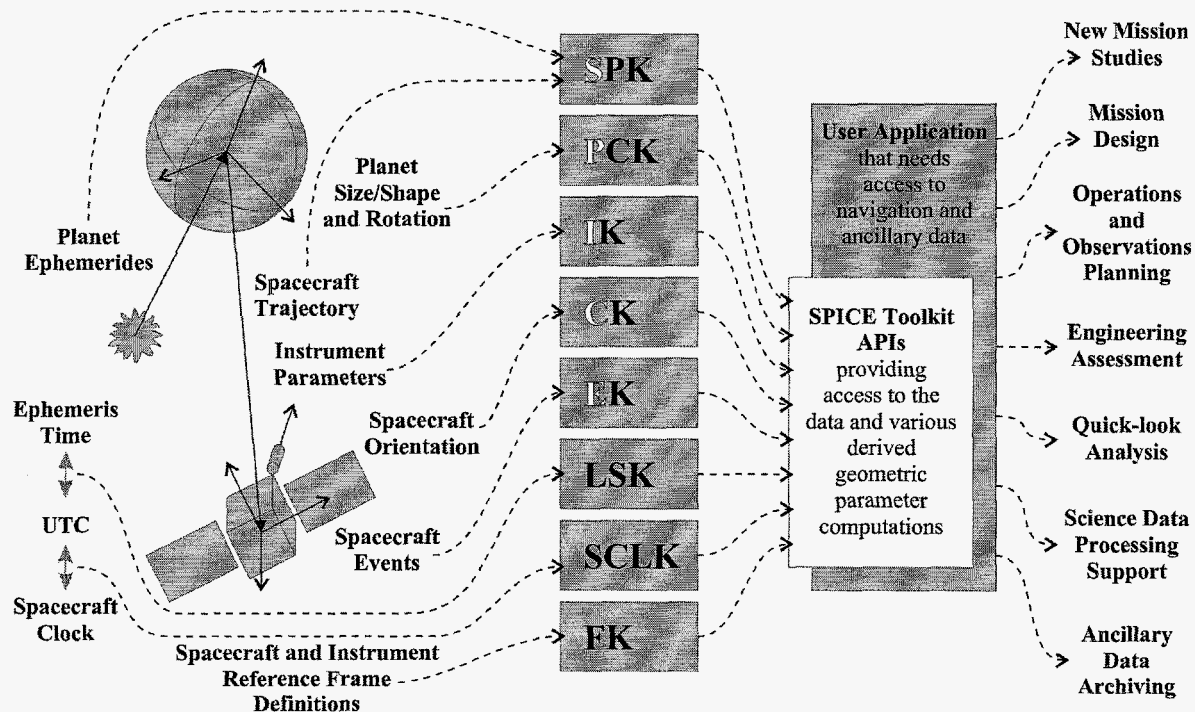


Diagram 1. SPICE Data and Software

grown steadily over its history is the fact that while SPICE evolved to satisfy ever increasing requirements of new missions and applications, it never turned away existing users by changing the functionality that was already in the system. Evolution is made possible by planning for extensibility of all major SPICE components, and the system as the whole. For example, while the first version of SPICE supported only three different representations of trajectory data, the current system supports eighteen, such as Chebychev polynomials, Lagrange and Hermit interpolation over discrete state vectors, and two line element sets. Neither small, such as addition of new SPK types, nor big updates, such as introduction of a major new subsystem, affected the way SPICE worked in existing applications at the time when the changes were made. The backward compatibility requirement that made this possible became policy in SPICE development from the very beginning and has been strictly adhered to throughout the history of the system. Any public SPICE APIs made available to users are guaranteed to be a part of the user interface forever, with their original functionality preserved or extended. The only exception to this policy is change in functionality due to bug fixes. The same backward compatibility policy is applied to the data representations: any kernel file that could be used with an earlier version of the system can be used in the same way with all future versions. This important trait of the SPICE system makes upgrading any SPICE-based application to a newer version of SPICE, often providing substantial additional capabilities, an easy task with very low costs associated with it.

From the very beginning SPICE developers recognized the diversity of the computing environments and programming languages that exist in the science and engineering communities of the space exploration domain. Currently SPICE is provided in three languages – FORTRAN, C, and IDL (Research Systems Incorporated Interactive Data Language) – for PCs running Windows, Linux, or Cygwin, Macs running OSX, and Sun and HP workstations (see Tab. 1). More than one compiler brand is supported for some of these environments. This multi-platform nature of SPICE facilitates its use by anyone in the space exploration community, including those who build multi-platform applications. Also, the fact that SPICE is available for less expensive computer types as well as that it is ported to a number of free operating systems and compilers allows for its use in ground systems based on less expensive computers, leading to substantial savings on required hardware and third party software.

Maintaining identical functionality in the set of high-level APIs implemented in different languages while providing APIs and documentation native for each of the languages was the main goal in the multi-language

Language	Hardware, OS, Compiler
FORTRAN	HP, HP-UX, HP Fortran
	Mac, OSX, Absoft Fortran
	Mac, OSX, g77
	PC, Windows/Cygwin, g77
	PC, Linux, g77
	PC, Windows, Digital Fortran
	PC, Windows, Lahey Fortran
	Sun, Solaris, Sun Workshop Fortran
	VAX, VMS, VAX Fortran
C	HP, HP-UX, HP C
	Mac, OSX, Apple C
	PC, Windows/Cygwin, gcc
	PC, Linux, gcc
	PC, Windows, MS Visual C
	Sun, Solaris, Sun C
	Sun, Solaris, gcc
IDL	Mac, OSX, Apple C, IDL 6.1
	PC, Linux, gcc, IDL 6.1
	PC, Windows, MS Visual C, IDL 6.1
	Sun, Solaris, Sun C, IDL 6.1
	Sun, Solaris, gcc, IDL 6.1

Table 1. Supported Languages and Environments

SPICE development. Since at the time when SPICE started FORTRAN was the language of choice in the science and engineering space community, the core of the SPICE system was developed and still exists in FORTRAN. As the needs of the user community evolved, the C language interface to SPICE, CSPICE, was introduced. To achieve the goal stated earlier, SPICE was not re-written in C at that time. Instead, FORTRAN SPICE source code was first converted to C using the public f2c conversion tool. Then a set of "wrappers" having interfaces natural to C programmers and calling the f2c'ed code was added. Finally a complete set of C specific documentation was added. A similar approach was taken with the recently added SPICE implementation in IDL, "ICY". It was developed as a set IDL routines calling CSPICE interfaces made available to IDL via a shared object library. The fact that all three implementation of SPICE – FORTRAN, C, and IDL – provide the same set of high level APIs and use the same underlying algorithms to access the data and perform computations guarantees that the same results will be returned to users no matter which language is chosen. For example, a heritage downlink scheduling tool calling FORTRAN SPICE, an attitude control analysis application linked to CSPICE, and a science observation planner's IDL script invoking ICY will compute the same spacecraft position for a given time when they use the same high level SPICE APIs and read the same SPICE data. While as of today SPICE is released only in these three languages, it is important to note that the CSPICE APIs can be and are called from other languages, such as C++, Java, Matlab, perl, python, and even MS Excel. The availability of SPICE in a few languages together with the possibility to call

SPICE APIs from many languages allows for great diversity in developing applications calling SPICE. This diversity often translates to greater efficiency in developing applications, which in turn leads to overall savings on tools development.

Data portability is another important aspect of multi-platform support that was addressed by SPICE from the very beginning. Initially it was done by employing conversion of SPICE's binary file types to a special transfer format so they could be moved to and converted back to binary format on a computer using a different binary architecture. While this mechanism is still supported today, a few years ago SPICE was augmented with the capability to read non-native binary files on big endian (Unix and Mac) and little endian (PC) platforms. This capability greatly simplifies SPICE data management for both data producers and users, allowing transparent access to SPICE binary data files on computers with incompatible binary architectures. This, for example, makes creating and maintaining large sets of SPICE data on a Unix server while accessing them via direct NFS mounts from a PC possible. Similar capability for text data files is planned to be added to the C and IDL implementations in the near future. (It is not feasible for the FORTRAN implementation of the system.)

SPICE is a true open source system as the SPICE distribution packages include the complete set of source modules for the main library and for the utility programs provided with the toolkit. An important attribute of the SPICE source code is that every module is written following strict programming style guidelines to ensure its readability and maintainability. In addition to that every module is extensively documented with the complete interface specification provided in the well-structured block of comments, called the SPICE module header, located at the top of the module. There are also extensive in-line comments interspersed with the actual code lines. Distribution of the well-documented source code makes algorithm implementations available to the users for inspection, thus eliminating the possibility of treating any part of the SPICE system as a "black box". This feature is desirable in any software system that is intended for use in critical operations.

In addition to SPICE headers that serve as the primary and most comprehensive interface specification for each of the SPICE APIs, a wealth of documentation describing the system on many different levels is distributed with the SPICE software. This documentation includes reference documents covering in detail all essential SPICE subsystems, user manual documents for all utilities provided with the toolkit, special index documents for quickly locating the APIs of interest by name or functionality, and toolkit

description documents. In addition to the documentation distributed with the toolkit, other types of informational aids are available on the NAIF web site. Among them are a comprehensive tutorial package comprised of about 40 presentations covering all aspects of SPICE, and a set of step-by-step programming lessons based on real mission data. These tutorials and lessons are intended for users just getting to know the SPICE system, but also benefit those who have been long time users since these packages are constantly improved and updated to reflect new functionality that becomes available in SPICE. Approximately twice a year NAIF conducts seminars and training sessions based on these packages. Most of these seminars are open to any current or potential SPICE users, both in the U.S. and in other countries.

Being a large software system – over 100,000 executable lines of code – SPICE has proven to be virtually bug-free, which is a crucial characteristic for any software used in space operations. The number of bugs discovered in the system is less than ten per year, with the majority of these bugs identified by the system developers themselves and affecting only obscure usage scenarios. This became possible due to the testing procedures to which the SPICE development team strictly adheres. Over the last ten years internal policies in the SPICE development team require a comprehensive set of test code be developed for and in parallel with every new family of SPICE code. This suite of test cases is used to verify all existing SPICE functionality, as well as any additions, on all of the supported computer environments when a new version of the system is prepared for release. The diversity of compiler brands and operating systems supported by SPICE has proven to be instrumental in identifying problems in the code prior to its release to the customer. The fact that none of the functionality available in the system was ever been retracted, combined with fact that it has been exercised in numerous applications on many flight projects, allows the NAIF team to rightfully call the system "tried-and-true".

4. SPICE Implementation for Space Missions

The capabilities of SPICE, built up as the result of constant development and application in over twenty space projects during the last one and half decades, are truly multi-mission and applicable in space projects of all types – flyby, orbiters, observatories, landers and rovers (see Tab. 2). Of the current U.S. planetary orbiter missions (Mars Global Surveyor, 2001 Mars Odyssey, Cassini) and flyby missions (Stardust, Deep Impact), SPICE is used in a variety of applications supporting observation planning, communication scheduling, pointing design and analysis, and science data analysis. Of the current U.S. rover missions (Mars Exploration Rovers) SPICE-based time conversion capabilities are

Restorations	Past Customers	Current Customers	Pending
Apollo 15, 16 [P]	Magellan [P]	Mars Global Surveyor	Phoenix
Mariner 9 [P]	Clementine (NRL)	Stardust	SIM
Mariner 10 [P]	Mars Observer	Cassini/Huygens	
Viking Orbiters [P]	Mars 96 (Russia)	2001 Mars Odyssey	
Pioneer 10/11 [P]	Hubble Telescope [S]	Spitzer [P]	
Halley Armada [P]	ISO [S]	Mars Exploration Rovers	Future Possibilities
Phobos 2 [P] (Russia)	MSTI-3 (ACT Corp.)	Mars Express (ESA)	NASA Mars Program
Ulysses [P]	OTD (MSFC)	Deep Impact	Discovery Program
Voyagers [P]	Mars Pathfinder	Mars Reconnaissance Orbiter	Explorers Program
	Mars Climate Orbiter	New Horizons/Pluto (APL)	New Frontiers Program
	Mars Polar Lander	Messenger (APL)	JIMO
	NEAR (APL)	Mars Science Laboratory	Lunar Orbiter 08
	Deep Space 1	Rosetta (ESA)	BepiColombo (ESA)
	CONTOUR	Venus Express (ESA)	
	Space VLBI [P]	SMART-1 (ESA)	
	Galileo	Hayabusa (JAXA)	
	Genesis	SELENE (JAXA)	

[P]=partial use of SPICE, [S]=special tools or services provided by NAIF

Table 2. Past, current, and future space missions using SPICE system.

used through the ground system in planning and data processing tools. Some U.S. space-based telescopes (Hubble, Spitzer) utilize SPICE to plan observations of Solar System bodies. Moreover, all but one current and recent U.S. planetary mission use SPICE kernels as the primary mechanism for storing and distributing spacecraft position information within the project.

Since geometry information is needed across all phases in the mission life cycle, SPICE can be successfully utilized during every one of these phases. While new mission studies can take great advantage of the wealth of ephemeris data for planet, satellite, and small body ephemerides available as SPICE kernels, more detailed mission design activities make use of SPICE to perform observation coverage and long-term telecom analyses. During mission implementation the full power of SPICE can be used virtually in all functions required for planning and carrying out operations, from tactical uplink and downlink scheduling, to supporting time conversions through the ground system, and from spacecraft performance analysis to computations of geometry parameters required to support science data archiving.

SPICE provides especially great benefits when it is chosen as a standard for exchanging navigation and ancillary data between various functions within a project, both "along" the project life-cycle timeline and "across" it, at any given point in project implementation. When SPICE is used for this purpose, trajectory, attitude and time correlation data provided in SPICE kernels are produced and delivered to the project by groups that possess expertise in a particular area, such as orbit determination or attitude reconstruction. Since SPICE provides substantial flexibility in the way data

are stored in SPICE kernels as well as transparency in regards to how this data is accessed, the data producers adjust the production to make sure the end users accessing data through SPICE get correct position, orientation, and time conversion information. For the end users getting the "right" information becomes simply a matter of picking up and using the "right" SPICE kernels, making this essentially a file management task easily solvable in modern automated data production and delivery environments. SPICE SPK files are used as a standard way of exchanging trajectory data on all JPL planetary missions, with navigation team producing SPK kernels and other project teams – from DSN tracking services to science instrument teams – using them in SPICE based applications to get spacecraft position and velocity information. This helps eliminate the risk of misinterpreting trajectory data.

As demonstrated by experience both inside and outside of the US, the initial adaptation of SPICE for the first mission or experiment in a series, or at a new agency, requires a modest effort. But this is greatly facilitated by available detailed documentation and multi-mission data production tools. Among the tools included in the SPICE toolkit and freely distributed from the NAIF server are programs for conversion of position and orientation data, provided in a variety of representations as simple text files, to the corresponding SPICE kernel files. In addition to these programs NAIF has, in a few cases, developed applications that provide similar conversions for more specialized formats such as those used by JPL navigation software and ESOC's Data Distribution System. Actual SPICE kernels from past and current missions, freely available from the NAIF server and NASA's Planetary Data System, serve as a template for creating SPICE kernels, especially for the

kernels types that contain constant information, such as PCK, FK, and IK.

The initial effort of setting up SPICE production and usage has always paid off because SPICE adaptation for subsequent missions/experiments has always been just a small fraction of the initial investment, with the majority of tools based on SPICE requiring no or very minor changes. For data producers the first time deployment of SPICE means setting up conversion of position, orientation, and time correlation data to SPICE kernels. Most frequently this is done by developing a set of easily automated processes, usually implemented by scripts, invoking existing SPICE data production tools and delivering SPICE kernels to one of the project data servers. It also means creating the time-invariant SPICE kernels (IK, FK, PCK) using SPICE and project documentation. Similar kernels for other missions often serve as examples. For data users, the first time deployment of SPICE means incorporating calls to relevant SPICE APIs into their applications and setting up a process to obtain SPICE kernels from the data server. For the next mission or experiment carried out by the same agency or set of teams, only minor configuration changes will be needed. This was one of the main reasons why the European Space Research and Technology Centre (ESTEC) decided to adapt SPICE for distribution of navigation data to support science data archiving on the Rosetta, Venus Express, and SMART-1 missions after its successful application for Mars Express. It took a very short time for ESTEC's science archiving support team to modify the SPICE production and distribution processes and infrastructure established previously.

5. Future SPICE Development

Throughout its history SPICE evolved in many different ways to meet the tactical and strategic needs of the space engineering and science community by adding new generic functionality, adapting additional representation of data in existing subsystems, providing implementation in other languages, and adding support for new computer environments. This evolution continues today with many on-going developments that will directly benefit current and future SPICE users. Among the new SPICE subsystems currently under development are a comprehensive event finding capability, representations for natural body shapes by digital terrain model and tessellated plate model, catalogue of sky objects, and ability to accumulate and access SPICE data in application memory rather than in kernel files. The SPICE development team is also working on adding MATLAB and Java Native Interfaces (JNI) interfaces to the system as well as experimenting with various client-server architectures that could provide network access to SPICE

functionality and the wealth of the data available on the NAIF server.

6. Conclusions

Using SPICE opens the way to substantial reductions in the cost of development of ground system software. The fact that SPICE provides a way to store and access all kinds of navigation and ancillary data makes it a "one-stop-shop" and eliminates expenses associated with maintaining multiple systems providing these kinds of data. Extensive derived geometry parameter computation capabilities available in SPICE lead to substantial savings by eliminating the need for developing software components implementing these computations in the tool sets used by various ground system functions. SPICE availability for less expensive computer types, operating systems, and compilers makes possible using these platforms throughout the ground system, which might translate to substantial saving in ground system hardware and third party software acquisitions. The possibility for users to access SPICE APIs from many different languages allows development to be done in the language of the user's choice, making it more efficient and leading to savings during the development cycle. Comprehensive documentation and access to the source code facilitate development and maintenance of SPICE-based user applications by making all aspects of the SPICE system transparent and eliminating any efforts that would have been lost if it was a "black box". Very high reliability of SPICE code, resulting from rigorous testing by NAIF and proven by everyday use by many active missions, eliminates any operations costs that could result from dealing with software bugs. The truly multi-mission nature of SPICE allows for cost savings by developing SPICE-based tools, both on data production and data usage sides, that are applicable to all types of space missions and are easily configurable for a new mission or experiment. Finally, SPICE software and data are free for individual users and are available "24/7" on the NAIF web and FTP servers.

More information about the SPICE system as well as the SPICE toolkit and SPICE data for many past and current planetary missions are available on the NAIF Web Server: <http://naif.jpl.nasa.gov>.

7. References

- [1] Acton, C. H. Ancillary data services of NASA's Navigation and Ancillary Information Facility. *Planet. Space Sci.*, Vol 44, No. 1, pp 65-70, 1996.

The work described in this paper has been conducted by the Jet Propulsion Laboratory, California Institute of Technology, under contract with the National Aeronautics and Space Administration.